

Vous peignez, je suppose

需求分析与概要设计

1. 项目说明

1.1. 项目目标：

本项目旨在开发一款具有创新性和趣味性的网页联机“你画我猜”游戏，为玩家提供独特且富有挑战性的互动娱乐体验。游戏不仅是简单的娱乐活动，更是玩家展示自我创意、锻炼思维能力的舞台。在游戏过程中，玩家凭借天马行空的想象力和独具匠心的绘画技巧，将脑海中的抽象概念转化为生动形象的画作，与其他玩家展开紧张刺激又妙趣横生的互动。每一次绘画都是一次创意的绽放，每一次猜测都是一次思维的碰撞，玩家在互动中建立起深厚的情感连接，享受无尽的欢乐时光。软件还将不断引入新元素，保持新鲜感和吸引力，持续丰富游戏内容，涵盖多种主题和场景，满足不同玩家的个性化需求，致力于成为玩家休闲娱乐的选择。

- 构建一个支持六人同时在线的网页游戏平台，打破地域和时间限制，设计多种社交互动机制，鼓励玩家之间相互竞争和合作。采用“五猜一画”的游戏模式，与传统的你画我猜游戏形成差异化竞争。每一轮游戏中，随机选择一名玩家作为画画者，其余五名玩家作为猜测者。这种模式增加了游戏的不确定性和策略性，画画者需要在有限的时间内用画笔准确传达词汇的含义，同时还要考虑如何迷惑猜测者；猜测者则需要仔细观察画作，发挥自己的想象力和推理能力，尽快猜出正确答案。
- 引入独特的得分规则：若只有不到四人猜对答案，画画者将获得得分奖励，但无法满分，促使画画者更加谨慎地选择绘画策略；若有四个人猜对答案，画画者将得满分，这鼓励画画者采用独特的绘画技巧和表达方式，增加游戏的趣味性和挑战性；若五个人全猜对答案，画画者不得分，这增加了游戏的悬念和戏剧性，让玩家在游戏过程中始终保持高度的紧张和专注。
- 建立一个庞大且多样化的词汇库，涵盖动物、植物、食物、交通工具、娱乐明星、历史事件等多个领域，满足不同玩家的兴趣爱好和知识背景。词汇库会定期更新，加入当下热门的话题和流行文化元素，确保游戏内容始终保持新鲜感和吸引力。

- 确保游戏在网页端的流畅运行，采用先进的前端技术和优化策略，减少页面加载时间和卡顿现象。支持多种主流浏览器，让玩家可以在不同的设备和浏览器上自由畅玩游戏。设计简洁美观、易于操作的游戏界面，注重用户体验和交互设计。界面布局合理，功能按钮清晰明了，玩家可以轻松找到自己需要的功能和信息。
- 提供详细的游戏教程和提示，帮助新手玩家快速上手游戏。对游戏进行严格的测试和优化，及时修复游戏中出现的漏洞和问题，确保游戏的稳定性和可靠性。关注用户反馈和建议，不断改进和完善游戏功能和玩法，提升用户满意度和忠诚度。

1.2. 软硬件环境需求

1.操作系统

Windows 10/11 (64 位)

2.开发工具与运行时环境

Node.js：用于前端构建、依赖管理和本地开发服务器。

Vue CLI：快速初始化 Vue 前端项目。

.NET SDK：支持 ASP.NET Core 后端开发。

MySQL：数据库服务，需安装 MySQL Server。

VS Code：轻量级 IDE，需安装相关扩展插件

3.网络要求

本地开发需开放以下端口：

前端：3000 (Vue 开发服务器)

后端：5000 或 8080 (ASP.NET Core Kestrel 服务器)

数据库：3306 (MySQL 默认端口)

生产环境需支持 HTTPS 协议，并确保 WebSocket 通信 (用于实时画板同步) 的稳定性。

4.浏览器兼容性

支持 Chrome、Edge、Firefox、Safari 等现代浏览器 (需兼容 WebSocket 和 HTML5 Canvas)。

1.3. 使用的关键技术：

1. 前端技术 (Vue + HTML5 Canvas)

- 响应式界面：基于 Vue 3 的 Composition API 实现数据驱动视图，结合 Vue Router 管理游戏页面路由 (如大厅、画板、猜词页)。
- 实时画板功能：

- 技术难点 :通过 HTML5 Canvas 实现低延迟绘图 ,需处理笔触坐标实时同步、多端分辨率适配。
- 解决方案 :使用 Fabric.js 库简化 Canvas 操作 ,结合 WebSocket 传输绘图数据 (如路径坐标、画笔粗细、颜色)。
- 实时通信 :通过 WebSocket(或 SignalR)同步玩家动作(如猜词、倒计时) ,确保多人交互的即时性。

2. 后端技术 (ASP.NET Core + C#)

- RESTful API 设计 :提供游戏房间管理、用户认证、词库获取等接口。
- 实时同步 :
 - 技术难点 :高并发下实时画板数据的广播效率。
 - 解决方案 :使用 ASP.NET Core SignalR 实现双向通信 ,优化消息分发的分组策略 (如按房间号分组)。
- JWT 认证 :保障用户登录与会话安全 ,防止未授权访问游戏房间。

3. 数据库设计 (MySQL)

- 数据结构优化 :用户表 (存储账号、分数)、游戏房间表 (状态、玩家列表)、词库表 (分类、难度分级)。
- 性能难点 :频繁读写操作 (如实时更新玩家分数)。
- 解决方案 :通过索引优化查询效率 ,使用 Redis 缓存热点数据 (如词库、排行榜)。

4. 跨平台与部署

- Docker 容器化 :将后端、数据库分别容器化 ,实现一键部署和水平扩展。
- Nginx 反向代理 :配置负载均衡和 HTTPS 证书 ,提升生产环境稳定性。

2. 需求分析

2.1. 系统用例

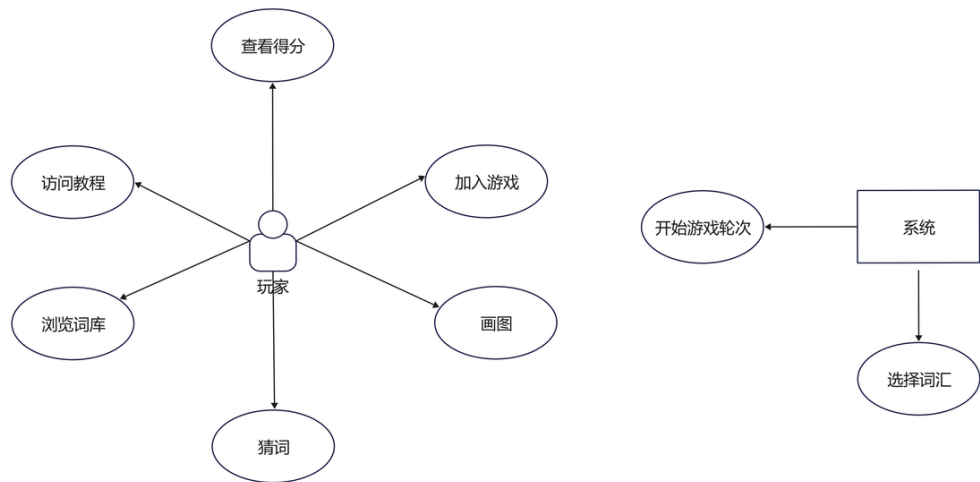


图 1 系统用例图

1. 加入游戏

参与者：玩家

描述：玩家进入游戏平台，选择或创建一个游戏会话并加入，最多六人后系统开始游戏。

基本事件流：玩家进入网页游戏大厅，选择一个现有房间或创建一个新房间，点击“加入”按钮进入房间。系统确认玩家全部准备，开始游戏轮次。

2. 画图

参与者：玩家（画图者）

描述：被指定为画画者的玩家收到系统提供的单词，并画图供他人猜测。

基本事件流：系统将玩家指定为本轮画画者，从词汇库中选择一个单词（通过“选择单词”）。玩家看到单词，使用网页绘图工具画图。画图实时显示给其他玩家（猜测者）。

3. 猜图

参与者：玩家（猜测者）

描述：猜测者查看画画者的图并提交对单词的猜测。

基本事件流：玩家在网页上看到画画者的图，在文本框中输入猜测并提交。系统

记录猜测，并在轮次结束时反馈（例如正确/错误）。

4. 查看得分

参与者：玩家

描述：轮次结束后，玩家查看得分（基于规则：4 人猜对画画者满分等）。

基本事件流：轮次结束，系统计算得分。玩家在网页上看到得分总结，包括个人得分和排行榜。

5. 访问教程

参与者：玩家

描述：玩家访问教程以了解游戏玩法和规则。

基本事件流：在游戏大厅，玩家点击“教程”按钮。系统显示教程页面，包含说明和示例。玩家浏览教程后返回大厅。

6. 浏览词汇库

参与者：玩家

描述：玩家浏览词汇库，查看可能出现的单词。

基本事件流：在游戏大厅，玩家点击“词汇库”按钮。系统显示分类单词列表，玩家浏览后返回大厅。

7. 选择词汇

参与者：系统

描述：系统从词汇库中随机选择一个单词给画画者。

基本事件流：系统访问词汇库，随机选择一个分类中的单词，将单词提供给画画者。

8. 开始游戏轮次

参与者：系统

描述：系统在玩家加入后启动游戏轮次。

基本事件流：系统确认准备玩家人数（最多六人）。随机指定一名玩家为画画者，其余为猜测者。开始轮次，通知所有玩家角色。

2.2. 业务流程

见 4

3. 概要设计

3.1. 功能模块设计

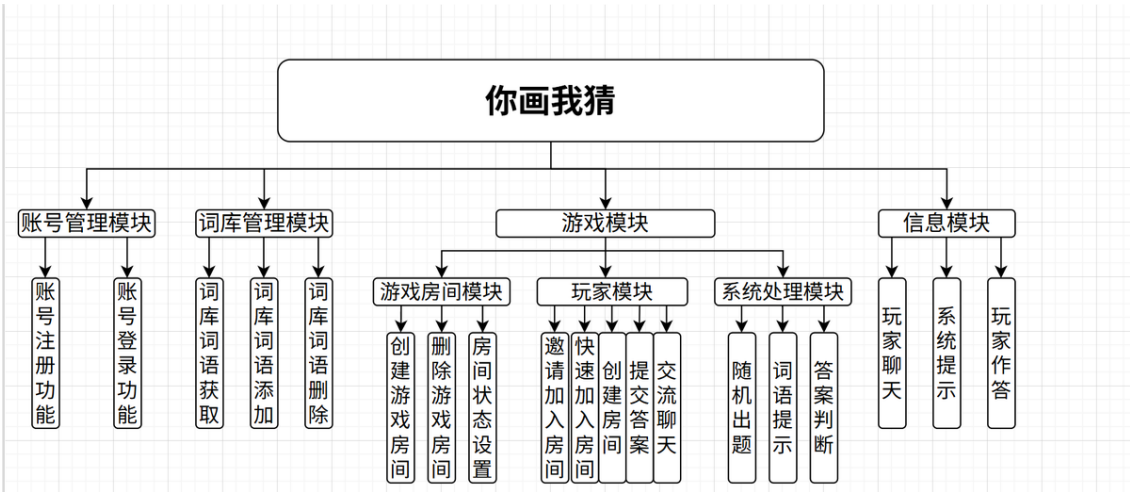


图 2 功能模块图

1.账号管理模块

输入：用户名、密码

输出：登录成功信息或失败提示、用户身份

功能概述：用于处理玩家的登录、注册、身份验证。用户登录后系统生成唯一身份标识，在整个游戏中进行用户信息记录和权限控制。

2.词库管理模块

输入：词语、词库操作指令（添加/获取/删除等）

输出：词库中词语列表、操作成功或失败提示

功能概述：管理游戏中的词语资源，支持管理员添加新词语并供游戏调用，也支持从词库中获取游戏词语，实现游戏内容的更新与维护

3.游戏房间管理

输入：创建/删除房间请求、房间状态更新指令

输出：房间列表、操作结果、房间状态信息

功能概述：提供玩家对局的虚拟房间环境，支持创建和删除游戏房间、设置房间状态（如准备中、游戏中、已结束），确保游戏流程有序进行

4.玩家管理

输入：玩家加入/退出房间请求、邀请玩家信息

输出：玩家加入状态、房间成员信息

功能概述：管理玩家与游戏房间的交互，支持玩家快速加入房间、接受邀请、退出游戏操作

5.玩家游戏模块

输入：玩家聊天/作答内容

输出：聊天发送成功/作答结果提示

功能概述：玩家的讨论区既用于交流，也用于玩家的作答，当输出的字数与题目字数一致时，视为玩家作答情况，其余视为玩家的聊天讨论

6.系统处理

输入：玩家提交的答案、绘图信息、词语

输出：判断结果，提示信息

功能概述：对玩家的答案进行判断，与词库中的目标词进行匹配，同时根据情况提供词语提示等功能

7.信息交互

输入：玩家聊天信息、系统提示指令

输出：聊天内容、系统提示内容

功能概述：提供游戏中的交流平台，支持玩家之间的实时聊天，并通过系统提示引导玩家操作或提供帮助，提示游戏互动性与用户体验

3.2. 核心类图

3.1.1 后端类图

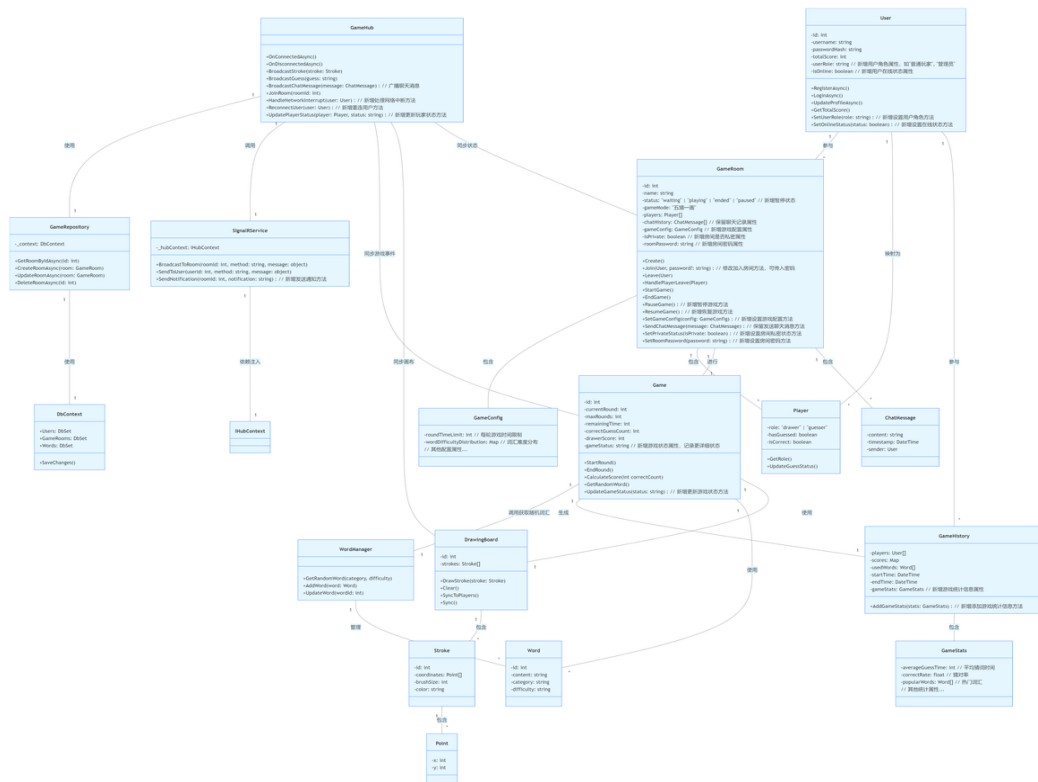


图 3 核心类图


1. `DbContext` 类

功能：作为数据库上下文，负责管理数据库连接和实体集合。它为应用程序与数据库之间提供了交互接口，是数据持久化的核心类。

职责：

持有 `Users`、`GameRooms` 和 `Words` 三个 `DbSet` 类型的属性,用于对数据库中对应的用户、游戏房间和词汇表进行增删改查操作。

提供 `SaveChanges` 方法，用于将内存中对实体的修改保存到数据库。

 后端类图.txt

2. `User` 类

功能：代表系统中的用户，存储用户的基本信息和操作方法。

职责：

包含用户的基本属性，如 `id`、`username`、`passwordHash`、`totalScore`、`userRole` 和 `isOnline`。

提供用户注册 `RegisterAsync`、登录 `LoginAsync`、更新个人资料 `UpdateProfileAsync`、获取总分数 `GetTotalScore`、设置用户角色 `SetUserRole` 和设置在线状态 `SetOnlineStatus` 等方法。

3. `Player` 类

功能：表示游戏中的玩家角色，记录玩家在游戏角色和猜测状态。

职责：

包含玩家在游戏角色 (`drawer` 或 `guesser`)、是否已经猜测 `hasGuessed` 和猜测是否正确 `isCorrect` 等属性。

提供获取角色 `GetRole` 和更新猜测状态 `UpdateGuessStatus` 的方法。

4. `GameRoom` 类

功能：管理游戏房间的创建、加入、离开、游戏控制等操作，同时存储房间的基本信息和状态。

职责：

存储房间的基本属性，如 `id`、`name`、`status`、`gameMode`、`players`、`chatHistory`、`gameConfig`、`isPrivate` 和 `roomPassword`。

提供创建房间 `Create`、玩家加入房间 `Join` (可传入密码以加入私密房间)、玩家离开房间 `Leave`、处理玩家离开 `HandlePlayerLeave`、开始游戏 `StartGame`、结束游戏 `EndGame`、暂停游戏 `PauseGame`、恢复游戏 `ResumeGame`、设置游戏配置 `SetGameConfig`、发送聊天消息 `SendChatMessage`、设置房间私密状态 `SetPrivateStatus` 和设置房间密码 `SetRoomPassword` 等方法。

5. `Game` 类

功能：负责游戏的流程控制和得分计算，记录游戏状态和数据。

职责：

包含游戏的基本属性，如 ``id``、``currentRound``、``maxRounds``、``remainingTime``、``correctGuessCount``、``drawerScore`` 和 ``gameStatus``。

提供开始一轮游戏 ``StartRound``、结束一轮游戏 ``EndRound``、计算得分 ``CalculateScore``、获取随机词汇 ``GetRandomWord`` 和更新游戏状态 ``UpdateGameStatus`` 等方法。

6. ``Word`` 类

功能：表示游戏中使用的词汇，存储词汇的基本信息。

职责：存储词汇的 ``id``、``content``、``category`` 和 ``difficulty`` 等属性。

7. ``WordManager`` 类

功能：管理游戏的词汇库，提供词汇的获取、添加和更新操作。

职责：

提供获取随机词汇 ``GetRandomWord``（根据类别和难度）、添加词汇 ``AddWord`` 和更新词汇 ``UpdateWord``（根据词汇 ``id``）等方法。

8. ``DrawingBoard`` 类

功能：代表游戏中的画板，管理画板上的绘图操作和数据同步。

职责：

包含画板的 ``id`` 和 ``strokes`` 等属性。

提供绘制笔触 ``DrawStroke``、清除画板 ``Clear``、将绘图数据同步给玩家 ``SyncToPlayers`` 和一般的同步操作 ``Sync`` 等方法。

9. ``Stroke`` 类

功能：表示画板上的一个笔触，存储笔触的详细信息。

职责：存储笔触的 ``id``、``coordinates``、``brushSize`` 和 ``color`` 等属性。

10. ``Point`` 类

功能：表示坐标点，用于记录笔触的坐标信息。

职责：存储坐标点的 ``x`` 和 ``y`` 坐标值。

11. ``ChatMessage`` 类

功能：表示游戏房间内的聊天消息，存储消息的相关信息。

职责：存储消息的 `content`、`timestamp` 和 `sender` 等属性。

12. `GameHistory` 类

功能：记录游戏的历史信息，用于后续的查询和统计分析。

职责：

存储参与游戏的玩家列表 `players`、玩家的得分 `scores`、游戏中使用的词汇列表 `usedWords`、游戏开始时间 `startTime`、游戏结束时间 `endTime` 和游戏统计信息 `gameStats`。

提供添加游戏统计信息 `AddGameStats` 的方法。

13. `GameStats` 类

功能：存储游戏的统计信息，用于分析游戏的表现和趋势。

职责：存储平均猜词时间 `averageGuessTime`、猜对率 `correctRate`、热门词汇 `popularWords` 等统计属性。

14. `GameConfig` 类

功能：存储游戏的配置信息，用于设置游戏的规则和参数。

职责：存储每轮游戏时间限制 `roundTimeLimit`、词汇难度分布 `wordDifficultyDistribution` 等配置属性。

15. `GameHub` 类

功能：处理游戏中的实时通信，实现玩家之间的操作同步和消息广播。

职责：

处理玩家连接 `OnConnectedAsync` 和断开连接 `OnDisconnectedAsync` 的操作。

广播绘图数据 `BroadcastStroke`、猜测信息 `BroadcastGuess` 和聊天消息 `BroadcastChatMessage`。

处理玩家加入房间 `JoinRoom` 的操作。

处理网络中断 `HandleNetworkInterrupt` 和用户重连 `ReconnectUser` 的操作。

更新玩家状态 `UpdatePlayerStatus`。

16. `GameRepository` 类

功能：负责与数据库进行交互，执行对游戏房间的增删改查操作。

职责：

依赖于 `DbContext` 类来访问数据库。

提供根据房间 `id` 获取房间信息 `GetRoomByIdAsync`、创建游戏房间 `CreateRoomAsync`、更新游戏房间信息 `UpdateRoomAsync` 和删除游戏房间 `DeleteRoomAsync` 等方法。

17. `SignalRService` 类

功能：封装了 SignalR 的服务功能，用于向游戏房间或特定用户发送消息。

职责：

依赖于 `IHubContext<GameHub>` 来实现消息的发送。

提供向指定房间广播消息 `BroadcastToRoom`、向指定用户发送消息 `SendToUser` 和发送通知 `SendNotification` 等方法。

18. `IHubContext<GameHub>` 类（在类图中以依赖注入的形式出现）

功能：作为 SignalR 的上下文，为 `SignalRService` 提供与客户端进行实时通信的能力。

职责：为 `SignalRService` 提供向客户端发送消息、调用客户端方法等功能，是实现实时通信的基础。

3.1.2 前端类图

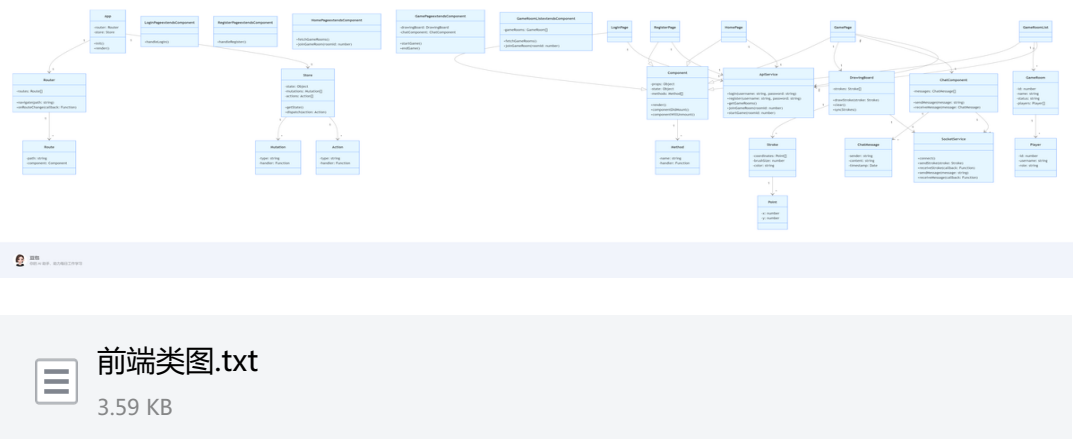


图 4 前端类图

1.核心架构类

App：作为应用的入口，负责初始化路由和状态管理，调用 `init` 方法启动应用，`render` 方法进行渲染。

Router：管理路由，根据路径导航到相应组件，`navigate` 方法用于导航，`onRouteChange` 监听路由变化。

Route：定义路由规则，包含路径和对应的组件。

Store：进行状态管理，存储应用状态，通过 `mutations` 修改状态，`actions` 触发 `mutations`。

Component：是所有页面组件的基类，有 `props`、`state` 和 `methods`，具备渲染和生命周期方法。

2.页面组件类

LoginPage：处理用户登录逻辑，`handleLogin` 方法发起登录请求。

RegisterPage：处理用户注册逻辑，`handleRegister` 方法发起注册请求。

HomePage：展示游戏房间列表，`fetchGameRooms` 获取房间列表，`joinGameRoom` 加入指定房间。

GamePage：游戏主页面，包含 `DrawingBoard` 和 `ChatComponent`，有 `startGame` 和 `endGame` 方法控制游戏流程。

3.游戏相关类

DrawingBoard：游戏画板，管理笔触，`drawStroke` 绘制笔触，`clear` 清空画板，`syncStrokes` 同步笔触。

Stroke：表示笔触，包含坐标、画笔大小和颜色。

Point：表示坐标点。

ChatComponent：聊天组件，管理聊天消息，`sendMessage` 发送消息，`receiveMessage` 接收消息。

ChatMessage：表示聊天消息，包含发送者、内容和时间戳。

GameRoomList：展示游戏房间列表，`fetchGameRooms` 获取房间列表，`joinGameRoom` 加入指定房间。

GameRoom：表示游戏房间，包含房间信息和玩家列表。

Player：表示玩家，包含玩家信息和角色。

4.服务类

ApiService：负责与后端 API 交互，提供登录、注册、获取房间列表、加入房间和开始游戏等方法。

SocketService：负责实时通信，提供连接、发送和接收笔触、消息等方法。

4. 界面设计

Vous peignez, je suppose

你画我猜页面流程图

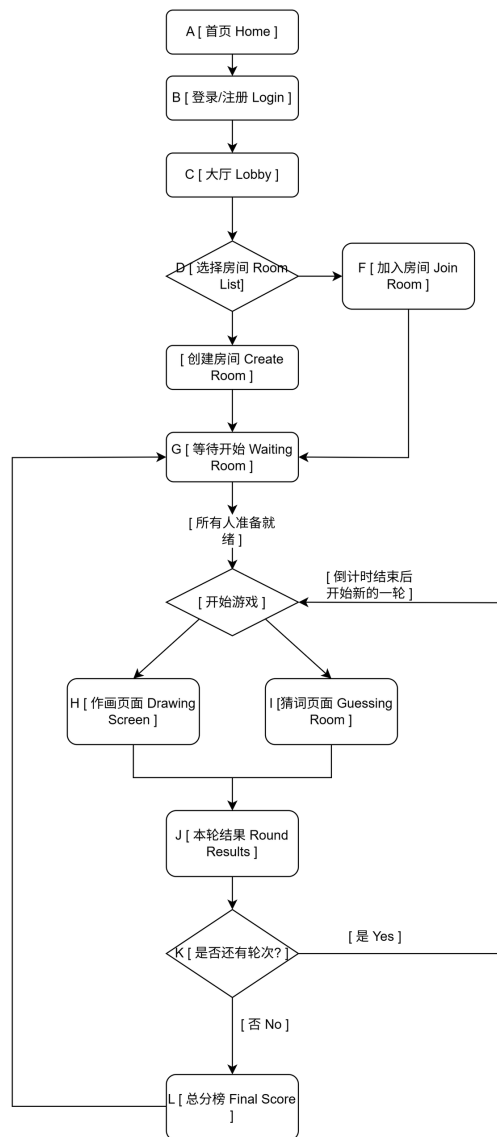


图 5 页面流程图

大厅		
[创建房间]	[刷新列表]	[设置]
房间列表:		
	房间 101 (3/6) 【加入】	
	房间 102 (6/6) 【已满】	
	房间 103 (2/6) 【加入】	
	底部: 昵称: _____->[修改昵称]	

图 6 大厅页面

创建新房间
房间名称: [_____输入名称_____]
最大玩家数: [4 ▼]
回合数: [6 ▼]
词库分类: <input type="checkbox"/> 动物 <input type="checkbox"/> 食物 <input type="checkbox"/> 日常 <input type="checkbox"/> 电影 <input type="checkbox"/> 地名 <input type="checkbox"/> 自定义
隐私设置: () 公开房间 () 私密房间 私密房间密码: [_____]
[创建房间] [取消返回]

图 7 创建新房间页面

等待开始		
房间号: 101 房主: Alice		
玩家列表:		
1. Alice (房主)		
2. Bob		
3. Charlie		
4. ...		
	[开始游戏]	[返回大厅]

图 8 房间内等待页面

当前回合：第 2 轮 Painter: Bob	
倒计时：00:45	
绘图区（画布）	
【画笔工具栏：画笔/橡皮/清除】	
猜词区域： > _____ [发送]	
底部列表： - Alice: apple (✗) - Charlie: banana (✗)	

图 9 画图和答题页面

回合结果
本轮词语： “apple”
[画图展示区]
猜对顺序：
1. Charlie (10 分)
2. Alice (8 分)
Painter Bob 获得：6 分
[倒计时 __s]

图 10 回合结束倒计时页面

欢迎来到 D&G	
[登录 Login] [注册 Sign Up]	
(登录)	
用户名/邮箱:	<input type="text"/>
密 码:	<input type="password"/>
<input type="checkbox"/> 记住我	
[登录]	
忘记密码? 立刻重置	
(第三方登录)	
[微信] [QQ] [微博]	

切换到 “注册” 栏:

创建新账号	
昵称:	<input type="text"/>
邮箱:	<input type="text"/>
密 码:	<input type="password"/>
确认密码:	<input type="password"/>
[注册并登录]	
已有账号? 去登录 →	

图 11 登录注册页面

设置	
个人信息	
• 昵称: Alice	[编辑]
• 头像: ●●●●●●	[更换]
通知与声音	
<input type="radio"/> 接收新消息提醒	
<input type="radio"/> 画画倒计时声音	
<input type="radio"/> 猜词提示音	
隐私与安全	
• 自动加入公开房间:	[开/关]
• 密码修改	[修改密码]
语言和地区	
• 语言:	[简体中文 ▼]
• 时区:	[Asia/Taipei ▼]
关于与帮助	
• 使用指南	[查看]
• 联系客服	[反馈问题]
• 版本号:	1.0.0
[保存设置] [重置默认]	

图 12 设置页面

总分榜			
本局房间号: 1234 共 N 轮: 4			
名次	玩家昵称	头像	总分
1	Charlie	(●●●●)	42 分
2	Alice	(●●●●)	38 分
3	Bob	(●●●●)	25 分
4	David	(●●●●)	18 分
...			
奖励: ★ 第一名奖励称号 “绘王”			
[返回房间]			

图 13 总分榜页面